

AN14849

i.MX RT700 Smartwatch Demo with Face Detection

Rev. 1.0 — 10 November 2025

Application note

Document information

Information	Content
Keywords	AN14849, i.MX RT700 MCU, smartwatch demo, face detection, GUI Guider tool, IAR workbench, Light and Versatile Graphics Library (LVGL)
Abstract	This document describes how to create a smartwatch UI using the i.MX RT700 MCU and the LVGL library. It includes a sample Smartwatch demo with face detection.



1 Introduction

The i.MX RT700 features up to five computing cores designed to power smart AI-enabled edge devices such as wearables, consumer medical, smart home, and HMI devices. Its Compute Subsystem includes a primary Arm Cortex-M33 running at 325 MHz. It also includes a 2.5D GPU with vector graphics acceleration and frame buffer compression. It incorporates the NXP eIQ Neutron NPU that accelerates AI workloads by up to 172x and integrates up to 7.5 MB of onboard SRAM.

This application note describes a smartwatch demo implemented on the MIMXRT700-EVK board. It describes steps to use the Light and Versatile Graphics Library (LVGL graphics library) to create a smartwatch UI on the board. It also includes a face recognition model that helps as a blocking screen. Once a face is detected, it activates a smartwatch, which displays a series of screens the user can interact with. The smart watch interacts with screens provided by applications such as an analog clock, a weather application, a fitness monitor, and a music player.

2 Graphics library and generate UI tool

This section provides an overview of the LVGL library and the GUI Guider (or generate UI tool).

2.1 LVGL overview

LVGL is an open source graphics library that is used to create embedded GUIs with easy-to-use graphical elements, beautiful visual effects, and a low memory footprint. The key features of this library are listed below:

- Open source and free to use under MIT license
- Written in C (C++ compatible) and hosted on GitHub
- More than 30 powerful, fully customizable widgets (Examples include the following: button, image button, checkbox, switch, slider, label, arc, bar, line, canvas, image, roller, slider, meter, table, text area, animation, calendar, chart, list, menu, message box, and tab view)
- Display of any resolution, GPU support, multidisplay support
- Supports multiple types of input devices.
- Drawing features
- Text features
- Image features
- Styles
- Micropython support
- Rich demo examples and documents
- Supported by GUI Guider, the free UI design tool of NXP

2.2 GUI Guider tool overview

The GUI Guider tool is a user-friendly GUI development tool developed by NXP. The tool enables the rapid development of high-quality displays using the open source LVGL graphics library. The drag-and-drop editor of this tool makes it easy to use the many features of LVGL. Widgets, animations, and styles can be used to create a GUI with minimal or no coding at all.

With the click of a button, you can run your application in a simulated environment or export it to a target project. The code generated from GUI Guider can easily be added to the MCUXpresso IDE, IAR Embedded Workbench, or to a Keil uVision project. It accelerates the development process and allows you to add an embedded user interface to your application seamlessly.

The GUI Guider tool is free to use with general purpose and crossover MCUs of NXP. It includes built-in project templates for several supported platforms.

3 System details

This section describes the details for the MIPI DSI and D-PHY clocks.

3.1 MIPI DSI and PHY

The MIPI DSI module in the i.MX RT700 MCU contains the MIPI DSI controller and D-PHY. It provides a high-speed serial interface between the host processor and the display module. The module implements all the layers and protocol functions defined in the MIPI DSI specification.

3.1.1 DSI host clock

[Table 1](#) describes the clocks that the DSI Host Controller requires.

Table 1. DSI Host Clock

Clock type	Clock alias	Description
reference clock	ref_clk	Input clock of the D-PHY PLL module that generates the High-Speed MIPI clock and High-Speed Data Lane signals.
Byte clock	clk_byte	The DPHY DLL module generates this clock, which is used for packet generation and packet transfer to the DPHY.
LP mode clock	clk_tx_esc	This clock is used by internal LPDT state machines and low-power transmission.
	clk_rx_esc	

3.1.2 D-PHY clock

The clock bus and data bus of D-PHY are differential structures. It supports two transmission modes, namely High-speed (HS) and Low-power (LP) mode. [Table 2](#) describes the application scenarios of different modes.

Table 2. D-PHY transmission mode

Transmission mode	Application scenarios
HS mode	High-speed synchronous data transmission
LP mode	Control command transmission
	Low-speed asynchronous data transmission

Table 3. D-PHY clock description

Clock type	Description
clk_byte	HS mode byte transmission clock. All data lanes share this signal.
clk_tx_esc	Transmit clock in LP mode, frequency varies from 12 MHz to 20 MHz
clk_rx_esc	Receive clock in LP mode, maximum frequency is 60 MHz

3.1.3 MIPI DSI and PHY clock distribution example

The demo described in this document uses the FRO0 running at 300 MHz as a clock source.

Example 1 shows the results of each clock distribution with FRO0 as the clock source.

- DPHY PLL (Input clock) = 300 MHz
- BitClock = 300 MHz
- RxClkEsc = 60 MHz (FRO0 / 5)
- TxClkEsc = 15 MHz (FRO0 / 5 / 4)

Example 1: FRO0 as clock source for MIPI DSI and PHY

```
CLOCK_AttachClk(kFRO0_DIV1_to_MIPI_DPHYESC_CLK);

/* RxClkEsc = 300 MHz / 5 = 60 MHz. */
CLOCK_SetClkDiv(kCLOCK_DivDphyEscRxClk, 5);
/* TxClkEsc = 300 MHz / 5 / 4 = 15MHz. */
CLOCK_SetClkDiv(kCLOCK_DivDphyEscTxClk, 4);
mipiDsiTxEscClkFreq_Hz = CLOCK_GetMipiDphyEscTxClkFreq();

CLOCK_AttachClk(kFRO0_DIV1_to_MIPI_DSI_HOST_PHY);
CLOCK_SetClkDiv(kCLOCK_DivDphyClk, 1);
```

3.2 LCD interface (LCDIF)

LCDIF is a high-performance, low-power, display processor unit (DPU). It outputs video and graphic streams from a single output panel. The single output panel can connect to either a display pixel interface (DPI) device or a display bus interface (DBI) device to support RGB output.

3.2.1 Display Bus Interface (DBI)

DBI-2 is a MIPI Alliance standard for parallel interfaces to display modules having display controllers and frame buffers. For systems based on these standards, the host processor loads images to the on-panel frame buffer through the display processor. Once loaded, the display controller manages the display and refresh functions on the display module without further intervention from the host processor. Image updates require the host processor to write new data into the frame buffer.

The signals between a host processor and the display module are listed below:

- **CSX** is the chip select signal for DBI Type A or B. The display module is selected when this signal is low.
- **Data [15:0]**: the host processor transfers each pixel value to the display module via the parallel data bus.
- **OE** is the data output enable signal.
- **D/CX** is the Data/Command indicator. The command is active when the signal is low.
- **R/WX for type A or RDX for type B**:
 - For Type A, specifies whether the current cycle is for read (1) or write (0) and set to 0.
 - For Type B, specifies whether the current cycle is for read (0) or write (1) and set to 1.
- **WRX** is the write signal for the DBI type B. The host processor writes data at the falling edge of the clock.
- **E** represents the E clock for the DBI type A. It is set to 1 for DBI A in fixed E mode.

This demo uses the DBI type B interface.

3.2.2 LCDIF DBI clock

When using the LCDIF, each interface (DBI or DPI) resides in different clock domains. When using DBI, the Media Main clock is used. For this example, the FRO0 is attached to the Media Main clock and divided by two, so the DBI interface runs at 150 MHz.

Example 2: LCDIF DBI clock source

```
/* DBI uses MEDIA MAIN clock. Use FRO0 / 2 equal to 150 MHz*/  
CLOCK_SetClkDiv(kCLOCK_DivMediaMainClk, 2);  
CLOCK_AttachClk(kFRO0_DIV1_to_MEDIA_MAIN);
```

3.3 VGPU for graphics drawing

The i.MX RT700 MCU contains a 2D Vector Graphics Processing Unit (VGPU). To draw graphics, the CPU sends instructions to the GPU. After receiving the instructions, the GPU performs the corresponding operations. During the period, the CPU is suspended, waiting for the instruction to complete. Various instructions are encapsulated in the API.

The i.MX RT700 MCU supports the OpenVG and VGLite Graphics APIs. Among them, OpenVG provides low-level hardware acceleration interfaces based on standard protocols for vector graphics libraries, such as Flash and SVG. VGLite is a lightweight API whose feature set is smaller than the OpenVG and is designed for the MCU. It has a smaller memory footprint and lower CPU overhead. This demo uses the VGLite API to draw scalar and vector graphics.

For detailed content of VGLite API, refer to [i.MX RT VGLite API Reference Manual](#) and [VGLite Vector Graphics API reference manual](#).

Among them, the VGLite API has two main rendering function APIs as mentioned below:

- **Vg_lite_draw**: This function renders vector graphics primitives so that there is no distortion after zooming in. Graphic elements are objects, and each object is a self-contained entity with attributes, such as color, shape, outline, size, and screen position.
- **Vg_lite_blit**: This function renders bitmap or raster images that consist of pixels. These pixels can be sorted and colored in different ways to form a pattern.

3.3.1 VGPU clock source

The VGPU has its own functional clock residing on the CLKCTL4, which is dedicated to the media domain. It supports a maximum frequency of 325 MHz. To achieve this frequency, 1.1 V level on the VDD2 domain is required. This voltage can be supplied by the PCA9422 PMIC on the MIMXRT700-EVK board.

This demo uses the VGPU running at 300 MHz using the FRO0 as a clock source.

Example 3: VGPU clock source

```
CLOCK_AttachClk(kFRO0_DIV1_to_VGPU);  
CLOCK_SetClkDiv(kCLOCK_DivVgpuClk, 1U);
```

3.4 ZC143AC72MIPI display

The ZC143AC72MIPI module incorporates a display controller that contains local registers and an internal buffer. It is a 1.43-inch, circular AMOLED display with 466 * 466 pixels and a 1-lane MIPI interface. It uses the driver IC CO5300 and the touch panel IC, TMA525B.

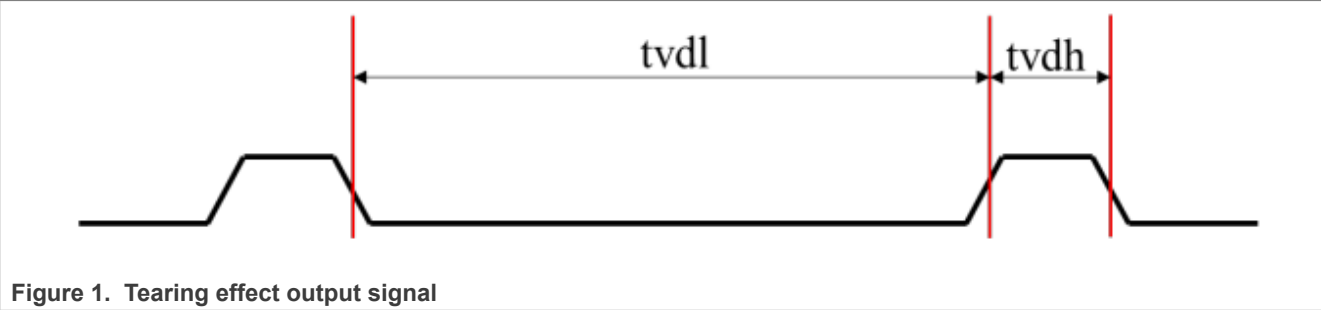
3.4.1 Tearing effect output signal

ZC143AC72MIPI module has a tearing effect output line (TE pin) to synchronize the MCU to frame writing. The screen refresh frequency refers to the frequency at which the device refreshes the screen, which is constant for a specific device. The refresh process is from left to right (row refresh), and then from top to bottom (column refresh). When completed, the signal can be issued to synchronize internal VSYNC.

The TELOM (Tearing effect line mode) has two types.

- Mode 1 refers to the tearing effect output signal consisting of V-sync information only.
- Mode 2 refers to the signal consisting of V-sync and H-sync information.

The `set_tear_off` (34h) and `set_tear_on` (35h) commands can be used to enable or disable these signals. The parameter of the `set_tear_on` (35h) and `set_tear_scanline` (44h) commands define the mode of the tearing effect signal. The tearing effect output signal is configured to Mode 1, as shown in [Figure 1](#).



The TE pin configuration is briefly summarized in [Table 4](#).

Table 4. TE pin settings instructions

Scenario	TE pin settings
1 frame send time < 1 frame refresh time	Set the interrupt at the start of VSYNC.
1 frame refresh time < 1 frame send time < 2 frame refresh time	Set the interrupt at the end of VSYNC.

In this demo, the specific situation is in line with scenario one. So, the TE pin can be set to rising edge trigger interrupt as shown in Example 4:

Example 4: TE pin setting

```
const gpio_pin_config_t tePinConfig = {
    .pinDirection = kGPIO_DigitalInput,
    .outputLogic = 0,
};

GPIO_PinInit(BOARD_MIPI_TE_GPIO, BOARD_MIPI_TE_PIN, &tePinConfig);

GPIO_SetPinInterruptConfig(BOARD_MIPI_TE_GPIO, BOARD_MIPI_TE_PIN,
kGPIO_InterruptRisingEdge);
GPIO_SetPinInterruptChannel(BOARD_MIPI_TE_GPIO, BOARD_MIPI_TE_PIN,
kGPIO_InterruptOutput0);

NVIC_SetPriority(BOARD_MIPI_TE_GPIO_IRQn, 3);
EnableIRQ(BOARD_MIPI_TE_GPIO_IRQn);
```

3.5 Camera interface

The i.MX RT700 does not contain a CSI camera interface module. As an alternative, the FlexIO is used to emulate the parallel camera interface to receive a real-time image from a camera device.

A parallel camera sensor normally has 8/10/16/24 data lines to output the pixel data, and an 8-bit interface camera is most commonly used. For such a camera, if a pixel is in an 8-bit size, one transfer cycle is required. If a pixel occupies more than 8 bits, more transfer cycles are required. For example, an RGB565 (16-bit) format pixel takes two transfer cycles, and an RGB888 (24-bit) format pixel takes three transfer cycles. Besides the data lines, there are timing control outputs VSYNC/VREF, HREF/HSYNC, and PCLK.

3.5.1 FlexIO clock source

Similar to the VGPU, the FlexIO has its own functional clock residing on the CLKCTL4. This demo also uses the FRO0 clock to run at 300 MHz.

Example 5: FlexIO clock source

```
CLOCK_AttachClk(kFRO0_DIV1_to_FLEXIO);  
CLOCK_SetClkDiv(kCLOCK_DivFlexioClk, 1U); /* FLEXIO 300MHz */
```

3.6 Neural Processor unit (NPU)

The NPU is an on-chip hardware accelerator that provides efficient processing of machine-learning workloads. It works with the CPU of the chip to deliver higher performance and lower power consumption for neural network (NN) inference.

In addition to managing Multiply Accumulate (MAC) compute blocks throughput, the NPU also performs the following functions:

- Optimizes arrangement of weights
- Manages memory accesses
- Provides rescaling, activation, and pooling functions

The eIQ Toolkit and eIQ Machine Learning (ML) Software Development Environment provide enablement support for the eIQ Neutron N3-64 NPU. Some of the features and operators listed in this section might not be enabled in the eIQ software. See [Section 8](#) for links to eIQ documentation that lists the features and operators enabled in the latest eIQ software.

You can import your trained and quantized Neural Networks (NN) into the [eIQ Toolkit](#). The eIQ toolkit generates an NPU-optimized model that can be integrated into an ML application running on the target device.

3.6.1 NPU clock source

The N3-64 NPU runs at the same clock frequency as the CPU0 as both share the same clock source. In the demo described in this document, the N3-64 NPU runs with FRO0 at 300 MHz.

4 Smartwatch demo overview

The smartwatch demo described in this document lets you explore the versatility of our smartwatch through an intuitive multiscreen experience. It helps the user to navigate seamlessly between multiple screens. For example, there can be screens displaying a classic analog clock face, simulating weather updates, and comprehensive health tracking including heart rate, steps, and activity levels. Each screen is designed for quick access and smooth interaction, which showcases the sleek interface and powerful functionality of the smartwatch.

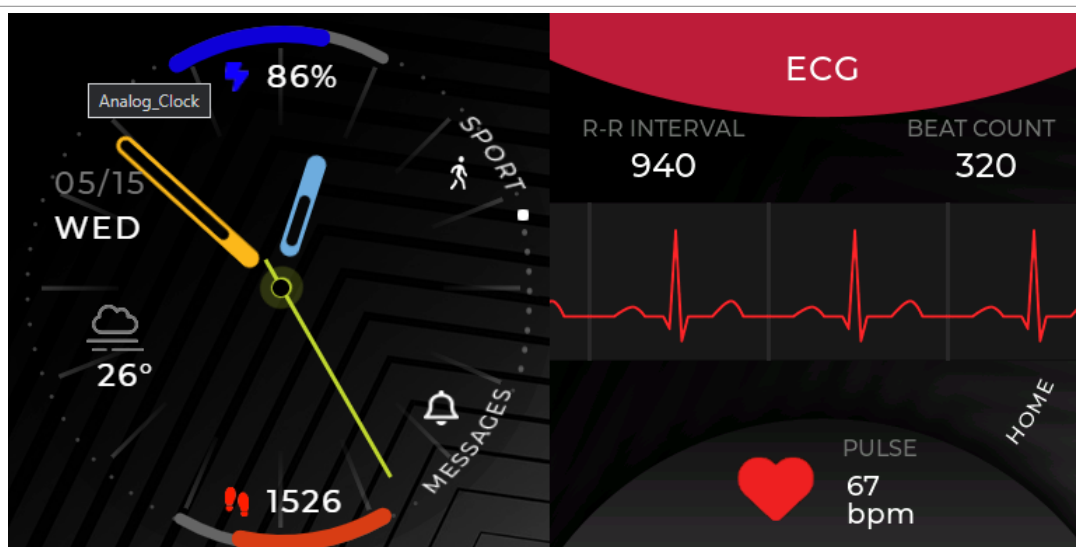


Figure 2. Smartwatch screen overview

The Face Detection Screen is a standout feature of this demo. It showcases real-time facial recognition powered by a machine learning model running directly on the MPU (Microprocessor Unit) of the device. This on-device intelligence ensures fast, private, and efficient face detection—demonstrating the edge AI capabilities of the smartwatch.

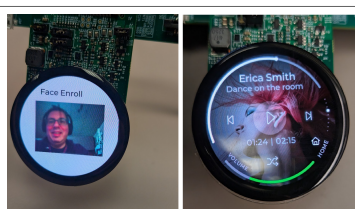


Figure 3. Face recognition screen

4.1 Program flow

1. Camera Data Acquisition

- A OV7670 camera module captures image frames.
- The FlexIO peripheral emulates the camera interface (parallel CSI) and receives the image data.
- Captured image data is transferred from FlexIO to system RAM using the EZHV engine for efficient memory handling.

2. Face Detection via ML Model

- The image data in RAM is passed to the NPU (Neural Processing Unit).
- The NPU runs a pre-trained face detection model to identify and locate faces in the frame.

- Inference results are stored or passed to the UI layer.

3. Parallel Image Display

- While the NPU processes the image, the same frame is sent to the LCD controller.
- The image is displayed on the screen via the LCDIF and MIPI DSI interface, enabling real-time visual feedback.

4. User Interface and Screen Navigation

- The LVGL graphics library manages UI rendering and screen transitions.
- When the user switches screens (for example, to clock, weather, or health apps), LVGL generates the corresponding frame buffer.
- The frame buffer is sent to the display through the LCDIF and MIPI DSI.

4.2 Memory structure

Memory management is one of the most critical aspects of an embedded device. The i.MX RT700 contains three XSPI interfaces to interact with external memories. XSPI0 is used for QSPI flash whereas XSPI1 is used for the PSRAM and is accessible by the CPU. XSPI2 is used with another PSRAM and is accessible by the media domain. In addition, it includes 7.5 MB of on-chip SRAM. [Table 5](#) serves as a reference to know where all the resources are located for the QSPI flash memory.

Table 5. QSPI flash memory overview

Resource	Address	Notes
Interrupt Vector Table	0x28004000 - 0x280042FF	-
Code and RO Data	0x28004300 - 0x29FFFFFF	-
UI Resource	0x2A000000 - 0x2AFFFFFF	Move to XSPI1 for fast CPU access
UI Resource	0x2B000000 - 0x2B7FFFFF	Move to XSPI2 for fast GPU access
Retina Model	0x2BC00000 - 0x2BDDFFFF	Move to SRAM for fast inference
Mobile Face Net Model	0x2BE00000 - 0x2BFFFFFF	Move to SRAM for fast inference

[Table 6](#) describes the internal SRAM memory.

Table 6. Internal SRAM memory resources

Resource	Address	Partition
FreeRTOS Heap - Cache	0x20000000 - 0x2007FFFF	SRAM P0
System Heap/Stack and RW Data, including 1 MB Heap specified for LVGL	0x20080000 - 0x201BFFFF	SRAM P1 – P11
Reserved for DMA descriptors	0x201C0000 - 0x201C3FFF	SRAM P11
NPU Model Memory	0x201C4000 - 0x204FFFFF	SRAM P11 – P15
Camera frame buffer A	0x20500000 - 0x2053FFFF	SRAM P16
Camera frame buffer B	0x20540000 - 0x2057FFFF	SRAM P17
CPU1 code	0x20580000 - 0x20587FFF	SRAM P18
Display frame buffer A	0x20588000 - 0x205FFFFF	SRAM P19 – P25
Display frame buffer B	0x20600000 - 0x2067FFFF	SRAM P26
GPU tessellation buffer	0x20680000 - 0x2077FFFF	SRAM P27 – P29

4.3 Running the demo

This section describes the hardware connections and steps to build the project. It also includes the steps to flash and run the demo.

4.3.1 Hardware connections

1. Plug in the OV7670 camera into connector **J53** as shown in [Figure 4](#).

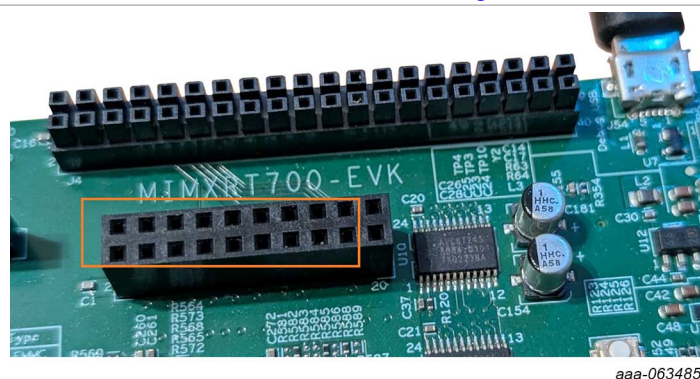


Figure 4. Connector J53 on the MIMXRT700-EVK board

While connecting, align the camera to the left-hand side so that the right-most column of pins (19 and 20) is left unconnected as shown in [Figure 5](#).

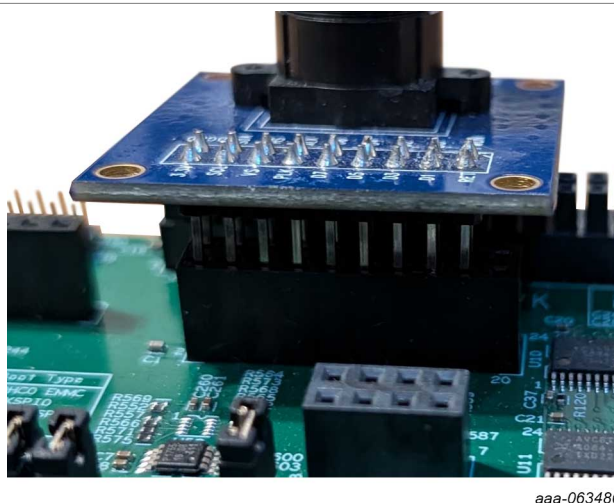


Figure 5. Camera connected to J53

2. Connect the ZC143AC72 LCD panel to the connector J26 as shown in [Figure 6](#).

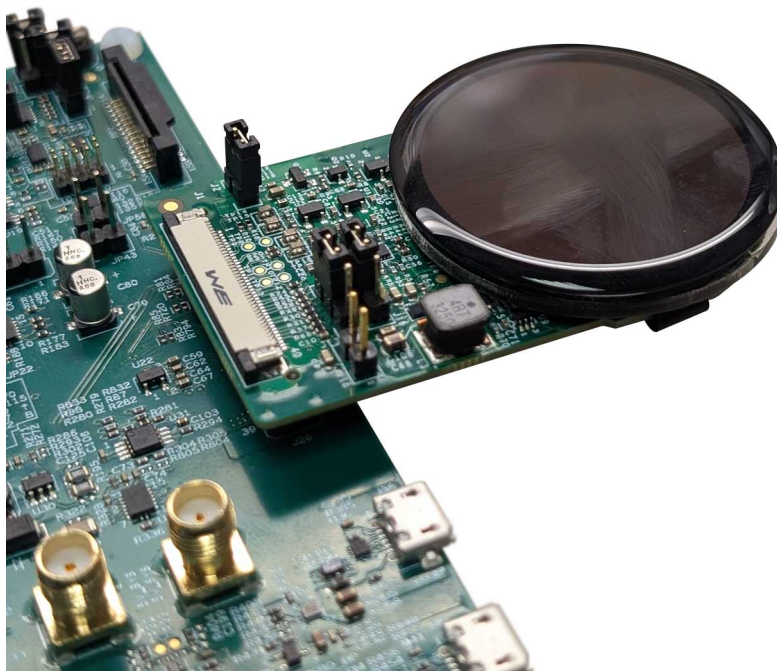


Figure 6. LCD connection to J26

4.3.2 Building the project

The software project requires IAR 9.60.2. Follow the steps below to build the project:

1. Open the IAR workspace at `\sdk\cm33_core1\power_mode_comp_only_cm33_core1.eww`.
2. Compile the project. This is the code for core1.
3. Now open the IAR workspace at `\sdk\cm33_core0\lvgl_guider.eww`.
4. Open the project settings by right clicking on the project name and selecting **Options...** as shown in [Figure 7](#).

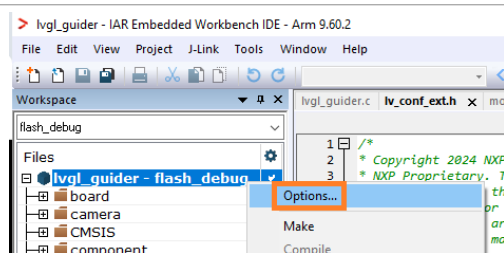


Figure 7. Project options

5. Go to the **C/C++ Compiler** category and select the **Preprocessor** tab. In the **Defined symbols** field, scroll down to the bottom and change the **FACE_MODEL_IN_FLASH** define to **0** as shown in [Figure 8](#).

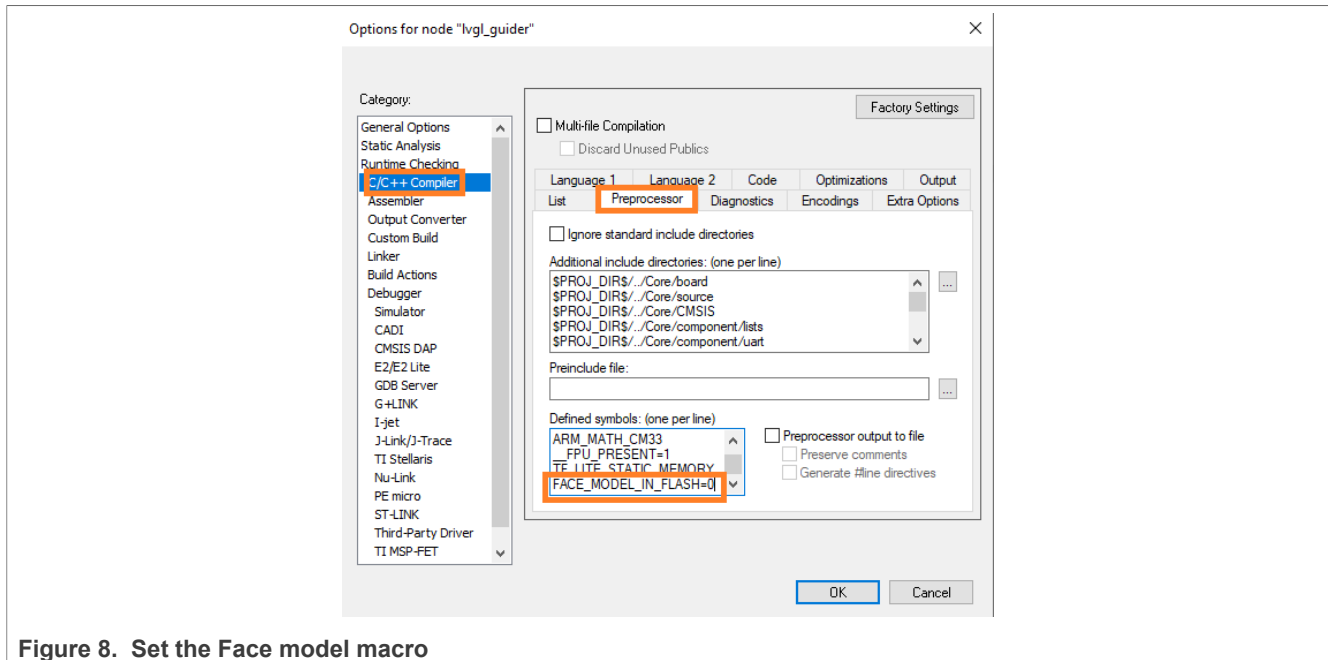


Figure 8. Set the Face model macro

6. Then, go to **custom\lv_conf_ext.h** and change **LVGL_UIRESOURCE_FIXADDRESS** to **0** as shown in [Figure 9](#).

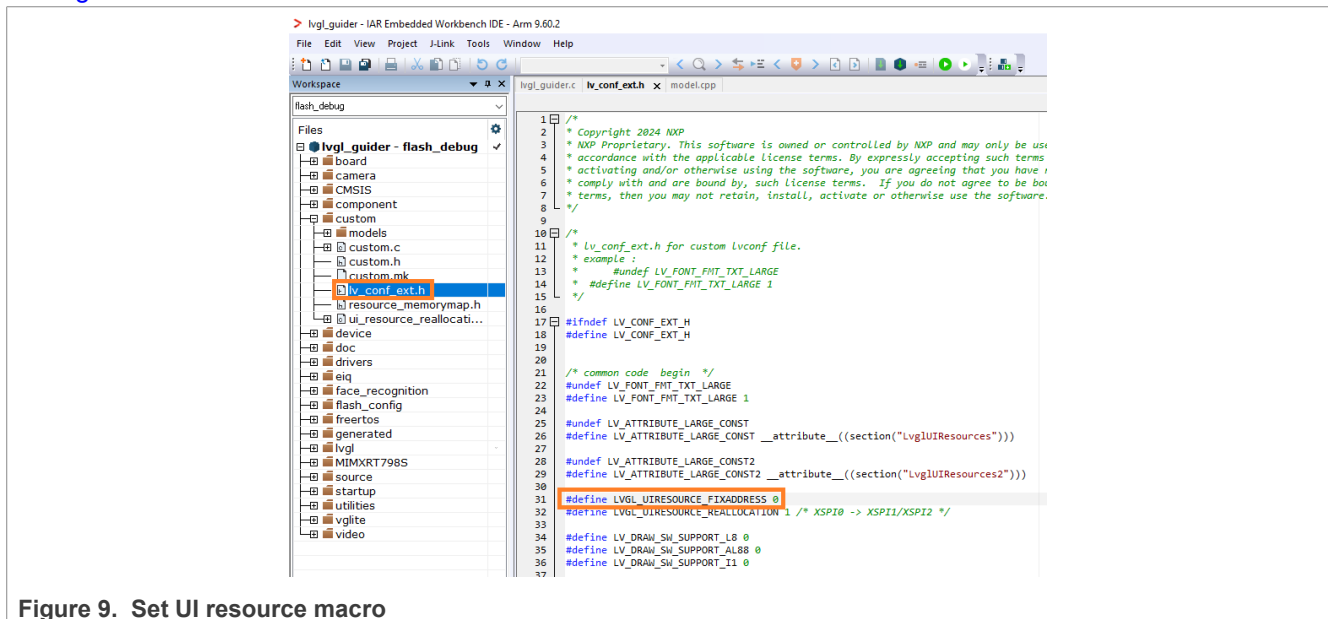


Figure 9. Set UI resource macro

7. Now compile the project.

4.3.3 Flashing and running the demo

1. On **J2** move the jumper to short the pins 1 and 2 if not done already.
2. Connect a 5V DC power cord to the MIMXRT700-EVK board at **J45**. An external supply must power up the board due to the LCD screen power requirements.
3. Slide the switch **SW4** to the ON position (5-6) as shown in [Figure 10](#).

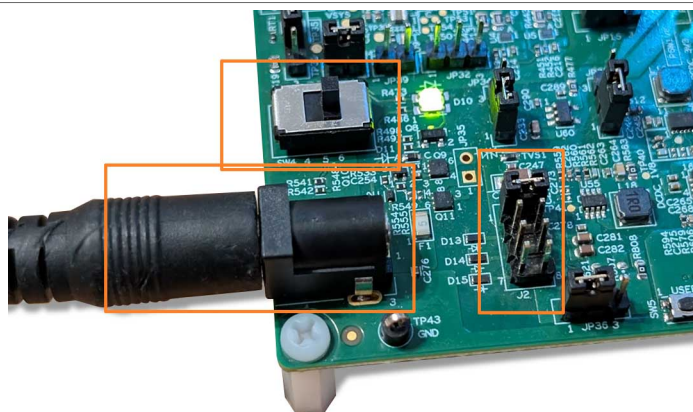


Figure 10. Power on the board

1. Connect a Micro B USB cable to **J54** at the top of the board, which is the onboard debugger connection. It can also be used as a COM port connection to read the messages printed to the serial terminal. However, the serial port is not required for this demo.
2. In the **IAR project settings**, set the debugger driver to the debug interface you are using on the board. Either **CMSIS DAP** or **J-Link** can be selected. By default, CMSIS DAP is selected.
3. Then, flash the project to the board by clicking the **Download and Debug** icon.
4. Now, a face recognition GUI is visible on the LCD screen. You can then swipe left and right to see other watch face screens or swipe down to go back to the face recognition GUI. See [Figure 11](#).



Figure 11. Running the demo

Attention: To modify the demo code, set the **LVGL_UIRESOURCE_FIXADDRESS** macro to **1** to avoid reflashing the GUI images again, the next time you flash the device. Setting this macro value significantly reduces the time taken to reflash the device after each code change. Only set this macro to **0**, the first time you program the flash to load the images into flash memory.

5 Performance evaluation

This section describes steps to measure the power consumed by different screens and to evaluate the result. The list of the screens is as below:

- Face detection screen.
- Analog clock: this clock updates the time every second as fast as it can and jumps to the Deep Sleep mode the rest of the time.
- Weather: this screen has an active animation, so the screen is updated continuously.

5.1 System power domain

The i.MX RT700 MCU has several power domains that are supplied by different power rails. In low-power applications, a power management IC (PMIC) is commonly used to supply power. On the MIMXRT700-EVK board, the PMIC PCA9422 supplies the power. Among these power domains, the significant ones are summarized in [Table 7](#).

Table 7. System power domain

Power domain	Power supply	Jumper where power can be measured
VDDN	PMIC_SW2	JP2 (pins 2 and 3 connected)
VDD2	PMIC_SW1	JP1
VDD1	PMIC_SW3	JP3
VDD1V8	PMIC_SW4	JP19
VDDIO	PMIC_SW4	JP26
VDDIO_2	PMIC_SW4	JP7 (pins 1 and 2 connected)
VDDIO_4	PMIC_SW4	JP61 (pins 1 and 2 connected)
VDDIO_5	PMIC_SW4	JP62 (pins 1 and 2 connected)
VDDIO_6	PMIC_SW4	JP63 (pins 1 and 2 connected)

5.2 Results

[Table 8](#) describes the power consumption analysis for the various screens used in the demo.

Table 8. Power measurement results

Power domain	Consumption (in mW)		
	Face detection	Analog clock	Weather
VDDN	11.26	11.26	11.26
VDD2	92	4.7	84
VDD1	0.028	0.028	0.028
VDD1V8	1.3	1.3	1.3
VDDIO	0.18	0.18	0.18
VDDIO_2	16.7	16.7	16.7
VDDIO_4	0.003	0.003	0.003
VDDIO_5	0.00071	0.0078	0.26
VDDIO_6	0.60	0.21	0.68

6 Conclusion

This smartwatch demo showcases the integration of advanced embedded technologies into a compact, power-efficient wearable platform. The system demonstrates flexible hardware interfacing capabilities by using a combination of display interfaces. The DBI and MIPI DSI are used for high-performance LCD control and the FlexIO is used to emulate the camera input. The demo also uses an on-device face detection model running on the MPU. This demo highlights the potential of edge AI for real-time, low-latency inference without reliance on cloud connectivity.

Power consumption analysis confirms that the system maintains efficient energy usage across all screens and processing modes, making it suitable for battery-constrained applications. Overall, this demo serves as a comprehensive reference for developers aiming to build feature-rich, AI-enabled wearables using modern microcontroller architectures and peripheral interfaces.

7 Acronyms

[Table 9](#) describes the acronyms listed in this document.

Table 9. Acronyms and abbreviations

Acronym	Description
AI	Artificial Intelligence
AMOLED	Active-Matrix Organic Light-emitting Diode
CMSIS	Cortex Microcontroller Software Interface Standard
DBI	Display Bus Interface
DSI	Display Serial Interface
EZHV	Enhanced Zero-copy Hardware Video
LVGL	Light and Versatile Graphics Library
LCD-IF	Liquid Crystal Display Interface
MAC	Multiply Accumulate
MIPI	Mobile Industry Processor Interface
ML	Machine Learning
NN	Neural Network
NPU	Neural Processing Unit
QSPI	Quad Serial Peripheral Interface
UI	User Interface
VGPU	Vector Graphics Processing Unit

8 References

[Table 10](#) lists some additional documents and resources that you can refer to for more information. Some of these documents are available only under a non-disclosure agreement (NDA). To access such a document, contact a local NXP field applications engineer (FAE) or sales representative.

Table 10. Related documentation

Document title	Description	Link/how to obtain
i.MX RT700 Reference Manual	Provides a detailed description about the i.MX RT700 MCU and its features, including memory maps, power supplies, and clocks	Contact your local NXP FAE/sales representative
i.MX RT700 Product Family Data Sheet	Provides information about the i.MX RT700 MCU electrical characteristics, hardware design considerations, and ordering information	
Learning VGLite API Programming on i.MX RT Series	Describes how to use the VGLite API for graphic programming with examples	AN14210
i.MX RT VGLite API Reference Manual	Describes the VGLite API, which can be used as the interface for the 2D GPU driver in NXP i.MX RT platforms.	i.MX RT VGLite API Reference Manual (document <i>IMXRTVGLITEAPIRM</i>)

For more documents, refer to the below URLs:

- <https://www.nxp.com/products/i.MX-RT700#documentation>
- <https://www.nxp.com/design/design-center/software/eiq-ai-development-environment:EIQ>
- <https://www.nxp.com/design/design-center/software/eiq-ai-development-environment/eiq-toolkit-for-end-to-end-model-development-and-deployment:EIQ-TOOLKIT#documentation>

9 Revision history

[Table 11](#) lists the revisions to this document.

Revision history

Document ID	Release date	Description
AN14849 v.1.0	10 November 2025	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

i.MX RT700 Smartwatch Demo with Face Detection

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

IAR — is a trademark of IAR Systems AB.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

Contents

1	Introduction	2
2	Graphics library and generate UI tool	3
2.1	LVGL overview	3
2.2	GUI Guider tool overview	3
3	System details	4
3.1	MIPI DSI and PHY	4
3.1.1	DSI host clock	4
3.1.2	D-PHY clock	4
3.1.3	MIPI DSI and PHY clock distribution example	5
3.2	LCD interface (LCDIF)	5
3.2.1	Display Bus Interface (DBI)	5
3.2.2	LCDIF DBI clock	6
3.3	VGPU for graphics drawing	6
3.3.1	VGPU clock source	6
3.4	ZC143AC72MIPI display	6
3.4.1	Tearing effect output signal	7
3.5	Camera interface	8
3.5.1	FlexIO clock source	8
3.6	Neural Processor unit (NPU)	8
3.6.1	NPU clock source	8
4	Smartwatch demo overview	9
4.1	Program flow	9
4.2	Memory structure	10
4.3	Running the demo	11
4.3.1	Hardware connections	11
4.3.2	Building the project	12
4.3.3	Flashing and running the demo	13
5	Performance evaluation	15
5.1	System power domain	15
5.2	Results	15
6	Conclusion	16
7	Acronyms	17
8	References	18
9	Revision history	19
	Legal information	20

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.